A Mobile Delay-tolerant Approach to Long-term Energy-efficient Underwater Sensor Networking

Eugenio Magistretti
DEIS
University of Bologna
emagistretti@deis.unibo.it

Jiejun Kong, Uichin Lee, Mario Gerla
Computer Science Dept.
UCLA
{jkong,uclee,gerla}@cs.ucla.edu

Paolo Bellavista, Antonio Corradi
DEIS
University of Bologna
{pbellavista,acorradi}@deis.unibo.it

Abstract— **Underwater environment represents a challenging and promising application scenario for sensor networks. Due to hard constraints imposed by acoustic communications and to high power consumption of acoustic modems, in Underwater Sensor Networks (USN) energy saving becomes even more critical than in traditional sensor networks. In this paper we propose Delay-tolerant Data Dolphin (DDD), an approach to apply delay-tolerant networking in the resource-constrained underwater environment. DDD exploits the mobility of a small number of capable collector nodes (namely dolphins) to harvest information sensed by low power sensor devices, while saving sensor battery power. DDD avoids energy-expensive multi-hop relaying by requiring sensors to perform only one-hop transmissions when a dolphin is within their transmission range. The paper presents simulation results to evaluate the effectiveness of randomly moving dolphins for data collection.**

I. INTRODUCTION

The vastness of the ocean, covering about two-third of the surface of earth, is still largely unexplored. This calls for the need of implementing unmanned underwater sensor network (USN) for periodic oceanographic monitoring. The new USN paradigm, however, poses formidable new challenges. Let us use the existing wireless radio sensor networks for a comparison. In contrast to wired networks, wireless radio networks operate in a resource constrained environment. Based upon technology for Dense Wave Division Multiplexing (DWDM), a single optical bundle can carry 12,800 GHz of optical signal. However, even the richest frequency band owner in the United States, namely Department of Defense (DoD), only owns approximately 300MHz of the total 3GHz of useable radio spectrum. Of the 300MHz owned by the military, individual systems are allocated in small blocks, e.g. 10MHz, 1MHz, or less. Consequently, protocols in wireless radio networking must be far more efficient than the same protocols in wired networks.

Nevertheless, if we extend our comparison to the underwater world, then wireless radio networks become the one with relatively much richer resource to expend. As high-frequency radio signals cannot propagate in water, underwater networking must rely on low-frequency acoustic communication, with the frequency upper bound reported as 1MHz at 60-meter range and reduced to the KHz level as the transmission range increases [10]. This implies that the entire acoustic band is less than several MHz and typical allocation is measured in KHz for individual systems. This drastic reduction in communication resource makes underwater networking an extremely challenging topic. The underwater wireless acoustic networks demand several orders of magnitude improvement in protocol efficiency compared to their "resource-abundant" wireless radio counterparts. In addition, acoustic communications impose far higher power consumption than RF based. The mechanical generation of acoustic waves, e.g., through ceramic projectors [6], makes modems power-hungry. Just to give some examples, the RF transmitter embedded in Crossbow Mica2 Motes [14], used for ground-based wireless sensor networks, consumes around 50mW to cover up to several tens of meters. State-of-the-art acoustic modems [6] [12] for USN drains 2 Watts to reach one to two-hundred meters.

In this paper we explore the power of mobility to alleviate the problems caused by the extremely constrained underwater communication. Based on the promising paradigm of delay-tolerant networking [5], we propose the design of *Delay-tolerant Data Dolphin* (DDD) to maximize the network lifetime of USN. DDD exploits mobile collectors, i.e., dolphins, scouring a region of interest where stationary sensors have been formerly deployed. Each underwater sensor is only required to transmit its data reports to the nearest dolphin in short one-hop distance, when the dolphin reaches its neighborhood. We have verified our design by simulation studies, investigating how parameter setting influences the effectiveness and efficiency of DDD. Since we consider the number of dolphins as a crucial design parameter, we have explored how the collection accuracy and the delay needed to harvest the reports depend on it.

The paper is organized as follows. Section II presents an overview of related work about delay-tolerant networking and underwater communications. Section III introduces DDD system architecture and protocol design. Then, Section IV presents our simulation results. Finally, Section V rapidly sketches plans for an analytic investigation of the problem, and Section VI concludes the paper.

II. RELATED WORK

DDD topic relates mainly to two different research areas, i.e., delay-tolerant and underwater networking. In the following, we discuss state-of-the-art contributions in these fields.

1525-3511/07/$25.00 ©2007 IEEE

A. Delay-tolerant networking

Many types of applications and challenging environments call for the introduction of the delay-tolerant architecture, a novel networking paradigm capable of operating in situations where simultaneous presence of senders and receivers cannot be assumed. Typical examples are exotic networks, e.g., deep space links [2], military networks, and underwater networks, where the limited transmission range and the mobility of the communication subjects prevent the establishment of persistent links.

These environments introduce critical aspects, hardly challenging traditional Internet architectures. In [5], Fall identifies a set of peculiar elements characterizing these types of networks: first of all disconnections, induced by mobility and energy conserving sleep states; in addition, high latency paths and low allowed data rates (due to radio and acoustic media). From a system point of view, interoperability and security represent critical challenges.

In order to face these challenges, a foundational architecture is proposed, called Delay Tolerant Networking (DTN) [5]. This aggregates heterogeneous networks, called *regions*, through DTN *gateways*. DTN gateways are in charge of moving messages from one of their adjacent regions to another, by physically transferring the information and by performing needed translations. Many challenges are identified, e.g., message routing, delivery reliability, retransmission management, security and authenticity, and flow/congestion control.

Research instances of delay-tolerant networking architectures have been proposed in the last years. [18] investigates epidemic protocols for the delivery of information about whales behavior to *Infostations*. Whales are equipped with sensing and communication devices. As soon as two whales are within mutual transmission ranges, they exchange carried data with a certain probability. When one whale reaches an *Infostation*, it offloads all gathered data. Differently from our proposal, [18] does not consider mobile collectors, but mobile sensing devices that epidemically spread information through other participants.

Another interesting example is provided in [21], where sparse networks are supported by *message ferries*. Remote node communities, without any chance of direct Internet connection, dispatch messages to ferries periodically passing in their neighborhood. Ferries provide communication services to the nodes, by delivering their messages to the Internet. Ferries are able to plan and modify their routes according to end user needs. Let us observe that in [21] remote communities are formed by mobile nodes and that ferry routes are not random, but can be planned proactively or reactively.

B. Underwater Networking

Some pioneering works on USN have been recently proposed. Some tutorial articles identify the main research challenges in this novel and uncharted area [8] [1] [4], by mainly concentrating on hardware and protocol issues. As deeply discussed in [16], underwater environment makes radio communications infeasible, due to the exceedingly high absorption rate, and calls for the deployment of acoustic modems. These devices are energy consuming, permit only low data rates, and impose an extremely high communication delay due to the low propagation speed of the acoustic waves. This effect highly challenges protocol design, and in particular medium access control solutions [16] [8] [4].

Other crucial services, like time synchronization and localization [8], call for novel research contributions to deal with the harsh environment. Networking and transport layers are not exempt from these issues [1]. A cross-layer design approach, including power management, coordination, and localization planes is highly advocated in [1]. [8] envisions interesting applications for untethered sensors deployed in the sea. [4] presents the interesting paradigm of mobile USN, i.e., networks where the underwater sensors move conveyed by water currents. Two different architecures are investigated, the one facing long-term non-time-critical applications (e.g., oceanography and pollution detection), the other short-term time-critical explorations (e.g., resource discovery and military surveillance).

Despite this significant work envisioning novel applications and high-level architectures, not much research has addressed the design of supporting communication protocols. Most original contributions have focused on acoustic modem design [6]. We propose DDD as one of the first efforts in the investigation of communication protocols specific for this uncharted field.

III. DELAY-TOLERANT DATA DOLPHINS

Many promising underwater applications such as seismic monitoring for oil extraction [8], ocean sampling and undersea exploration [1] call for the deployment of a large number of small stationary sensors. Then, sensed data can be collected in different ways: the traditional approach [16] is to recover the sensors after the task has ended; or alternatively, sensors can directly transmit data to remote stations (e.g., buoys). Both these solutions suffer from disadvantages. In the first case, time for data harvesting is forced to be the end of network lifetime. The system is inherently much more prone to unexpected inconsistencies or even collapses (due to lack of reconfiguration capabilities). In addition, regular nodes have to store the whole sensed data until they are recovered. In the second case, reports reach remote stations much faster, but they likely need to travel long paths, thus draining energy of nodes en route. Long-range communication also increases channel interference amongst different transmitters, in particular in the acoustic channel with very long propagation delay.

In this paper we study a delay-tolerant approach that permits to trade delay for energy consumption. DDD inherits advantages of both previously described approaches and, for instance, suits the periodic oceanographic monitoring applications, which can typically collect data periodically.

A. Architecture

In DDD, small stationary sensor devices are deployed on the seabed and continuously collect data from the environment. Data is locally processed, and is stored only upon the occurrence of interesting events (i.e., of special events that are designated by the command center). Beside sensors, a number of moving vehicles, namely *dolphins*[1], equipped with more capable devices scour the ground field to harvest data scattered over the network of sensors. Dolphins receive data packets only when they move within the one-hop neighborhood of related sensor nodes. Then, dolphins deliver gathered events as soon as they reach a remote base station.

This strategy permits sensors to spare the energy needed for long-range communications toward remote stations, at the price of a delayed delivery. Let us observe that considered applications are not time critical, thus do not impose real-time constraints on data delivery. However, compared to traditional solutions based on sensor recovery at the end of network lifetime, data harvesting delay is highly reduced. Simulation results will show that DDD system parameters, e.g., number of dolphins and buffer size, can be tuned to fit the delay requirements of the single application.

Sensor node assumptions. Sensors are energy and memory limited. To save energy, they spend most of time in a low-power sleep state, which reduces energy expense in various components like the sensor acoustic modem, the sensor sensing unit and the sensor mainboard. A sensor only periodically wakes up for environment sensing and event generation. Its communication device is an acoustic modem that includes two components [20]. One component mostly operates acoustic communications with near dolphin nodes. The other is a low-power transceiver used to determine the presence of dolphin nodes (by a special signal transmitted from dolphin nodes) and trigger the first one. Thus, the latter is turned on only when an event needs to be delivered, the former only when the first determines the presence of a dolphin. Sensors are equipped with a small amount of memory, so that they store a few recently-detected events. Due to memory constraint, events need to be replaced as memory fulfills.

In many applications, sensor data report must be tagged with location and time. We assume that each stationary sensor already knows its position. This can be the result of a careful deployment practice, or of the execution of a sensor localization protocol, which is a research challenge addressed in many research projects [8] for the harsh underwater environment.

Dolphin node assumptions. On the other hand, dolphin nodes are supposed to have large memory and energy conserves if compared with sensors. They collect and store events while moving in the field, and upload their memory to a near-shore base station for data delivery. Hence, they need to bear all events harvested between two consecutive passages to the base station (or a set of base stations in distributed deployment). In addition, they are equipped with powerful batteries, to overcome limitations on costly acoustic communication.

Dolphin node mobility will naturally be unplanned and unconstrained. Therefore, we study the random mobility case where dolphins move by following the random walk model; every dolphin node moves with an independently and identically distributed random pattern [11].

B. Protocol Design

To save energy, we propose that sensors exploit one-hop communications toward dolphins, i.e., that sensors deliver events only when a dolphin is within their sensing range. Once delivered, these events are stored in dolphin memory and can be safely removed from sensor memory.

Dolphins randomly move in the monitored area A, by periodically broadcasting beacons to advertise their presence. Beacons need to be transmitted on acoustic frequencies compatible with the low-power modems of the sensors. Advertising periods t are chosen according to sensor deployment and communication range r, and to the speed of the dolphins v. In this paper, we choose $t << \frac{r}{v}$ so that we can guarantee that almost every time a dolphin is within the communication range of a sensor, the sensor can realize its presence. This choice will drain more power from dolphin battery, but differently from sensors this has a much longer lifetime and can be easily replaced.

As soon as they determine the presence of a dolphin, sensors turn on acoustic modems and upload stored events to the dolphin. Since dolphin speed is low (on the order of few m/s for underwater motion), every time a sensor determines the presence of a dolphin, it will be able to deliver all stored events. Then, when dolphin moves in the communication range of a base station, it uploads all gathered events.

Dolphins will likely be costly nodes. As a consequence, our goal is to reduce the number of dolphins needed while serving the aquatic application requirements. If the number of deployed dolphins is not large enough, they will not be able to collect all needed event reports generated by the deployed sensors. Thus, our final goal is to determine the minimal number of dolphins that permits to statistically collect the requested large share of events. Since dolphins perform random motions and events are removed from sensor buffers when there is no space left, we do not expect to be always able to exhaustively collect all events. Instead, we want to investigate the probabilistic relation between the number of dolphins and the delivery ratio of sensor event reports.

IV. SIMULATION RESULTS

As a first effort in delay-tolerant networking for underwater scenarios, we simulated DDD on the well-known Qualnet simulator [17]. We modified Qualnet as in [19] to simulate underwater environments. We implemented a new medium and propagation model for underwater. In particular, the propagation speed of the communication signal was set to the speed of sound in water, i.e., *1500 m/s*. In addition, we used a bandwidth of 20 Kbps and a channel frequency of 50 KHz according to the values found in [10].

As a deployment scenario we considered a *2000m x 2000m* area, with 25 sensor devices, uniformly deployed on a *5x5* grid spanning the whole square (these measures are chosen according to the ranges given in [4]). We decided to model

---

[1] Let us remark that we do not just consider dolphins to carry the task, but any kind of fish sufficiently big to be equipped with a one-ounce device.

the mobility pattern of dolphins according to a general mathematical model, the random walk [11] [15], essentially based on brownian motion. Random walk logically splits time in identical slots; at the beginning of each slot, each dolphin chooses a constant random direction and follows that direction for the duration of the whole slot with a constant speed equal to *5m/s*. We realize that this assumption could be considered oversimplifying, but actually some recent research on fish movement presents similar models [9].

Let us observe that our goal is not to present precise results for the single examined application, but rather to show performance trends. We claim that by stretching or shrinking the deployment area, and varying dolphin speed or movement patterns, general trends will not be affected.

The results presented in the following refer to scenarios where sensors generate new events according to a Poisson distribution with average value *lambda*. Sensors store generated events in a local buffer with size *buf size*, until (a) the buffer is full (in that case we adopt a "least recently generated" replacement policy), or (b) the events have been uploaded to a dolphin (in that case, the buffer is emptied). Dolphins are equipped with an unbounded buffer. Where not differently specified, default values for *lambda* and *buf size* are chosen equal respectively to 1 event every *2 minutes* and *100* entries.

The goal of our simulations was to assess how the *Event Delivery Ratio* and *Delay* depend on three principal system parameters, i.e., the number of dolphins, the generation rate *lambda*, and the buffer size *buf size*. In more details, we define *Event Delivery Ratio* as the total number of events collected by the dolphins (i.e., the cardinality of the union of the events harvested by any single dolphin), out of the number of events generated in the area (i.e., the gross total of the number of events generated by any single sensor). The *Delay* is instead defined as the average delay intercurring between the time an event is generated and the time it is harvested for the first time by any dolphin (a similar metric is considered for the evaluation of packet delivery delay in MANET routing protocols [3]). All values presented in the following figures are the average of twenty independent simulation runnings. Obtained results showed a really low variance.

First, we investigate how the *Event Delivery Ratio* depends on the number of deployed dolphins and on the *buf size*.

Figure 1 shows the *Event Delivery Ratio* as a function of the number of dolphins (the plots refer to extreme *buf size* ranging from 1 to 500). As we expected, increasing the number of dolphins has a significant impact at first. As soon as a threshold value has been reached, the contribution of additional dolphins does not carry appreciable benefits. Let us rapidly note that the threshold value highly depends on event generation rate and buffer capacity of deployed sensors. As pointed out in Section III, DDD design does not address applications where an exhaustive collection of events is of vital importance. As a consequence, the number of dolphins suitable for the single case depends on application requirements.

Figure 2 investigates how the *Event Delivery Ratio* depends on the size of local buffers (the plots refer to 1, 3 and 7 dolphins cases). The growth is steeper in the first part of the graph, i.e., as the *buf size* increases from 1 to 100, then it slackens. This becomes evident as the number of dolphins increases: for the 7-dolphin case enlarging the buffer from 1 to 100 (resp. from 100 to 500) leads to a 56% increase (resp. 20%). In fact, by enlarging the *buf size* from 100 to 100+*x*, it is likely that, due to the high *Accuracy* of the multiple-dolphin case, most of the *x* additional entries harvested by a dolphin while in touch with a sensor have previously been collected by other dolphins, thus carrying no benefits.

Figure 3 shows how the generation rate *lambda* impacts on the *Event Delivery Ratio*. The plots show results for *lambda* ranging from 1 event every minute to 1 every 200 minutes, 1 and 7 dolphins, and *buf size* equal to 1 and 100. As expected, as the generation rate decreases (i.e., as *lambda* decreases) the percentage of harvested events increases. As the plot for 7 dolphins and *buf size* equal to 100 clearly shows, there exists a lower bound after which there is no substantial gain on *Event Delivery Ratio*. When a certain parameter setting leads to an *Event Delivery Ratio* approaching 1, the only events that are not collected at any moment are the ones recently generated. In fact, it is not feasible to obtain an *Event Delivery Ratio* equal to 1 because, as soon as dolphins move in a region, new events could be generated in uncovered areas in the latest seconds. The number of these events has a very small influence on the gross total and mainly depends on *lambda* and dolphin speed.

Figures 4 and 5 show how the *Delay* depends on the number of dolphins and on the *buf size* for two extreme values of *lambda*, 1 event every resp. 2 and 200 minutes. Both figures present the *normalized Delay*, i.e., the product between the *Delay* and the generation rate *lambda*. As the number of dolphins increases, the *normalized Delay* decreases, since events can be harvested more promptly by deploying a higher number of dolphins. As the *buf size* increases, the average harvesting delay increases. In fact, growing the *buf size* helps delivering to the dolphins more events, which have been generated less recently. This contributes improving the *Event Delivery Ratio*, but on the other hand, introduces older elements in the calculation of the *Delay*. Figure 5 shows that this holds until a threshold value is reached (dependent on the generation rate *lambda*). Let us consider the plots corresponding to *buf size* equal to 10 and 100. Due to the low generation rate, sensors are visited with a frequency that impedes they fulfill their buffer between two subsequent visits in either case. Thus, the *normalized Delay* tends to converge to the values of the plots corresponding to smaller *buf size*.

A crucial performance indicator is the time the dolphins need to harvest the events generated by a sensor. Obviously, since dolphins move randomly, the highest possible delay needed to get in touch with a chosen sensor is unbounded. However, we can experimentally determine an average measure of that interval. To this end, we set up a simulation where each sensor generated an event at time $T = 0$ and we measured the time needed to collect the events depending on the number of dolphins. Figure 6 reports on the y-axis the ratio of the sensors (out of the 25 deployed) visited after a certain time by the dolphins. Plots are averaged over 100 simulations and refer to the cases of 1, 3 and 7 dolphins. We observe that this final plot does not add qualitative information to the other figures, but only provides a tool for network designers to choose the number of dolphins they need to deploy depending on application-specific deadline requirements.

Only recently, we started measuring the energy-efficiency of our protocol. First, we are carefully tuning the Qualnet custom underwater physical layer to properly implement the energy consumption models of state-of-the-art modems available on the market [12]. Second, we are implementing alternative solutions, as described in Section III; two extreme cases are under consideration: the first one is trivially based on the collection of all sensors at the end of the task; the second includes the deployment of one or more stationary base stations (buoys), spread on the sensing field, and receiving data from sensors as soon as new events are generated. We claim that outcomes of these experiments are qualitatively straightforward; nonetheless, actual simulations will provide insights on how different parameter values (e.g., number of buoys) precisely influence the results. This can help application designers to choose the solution that best suits their case.

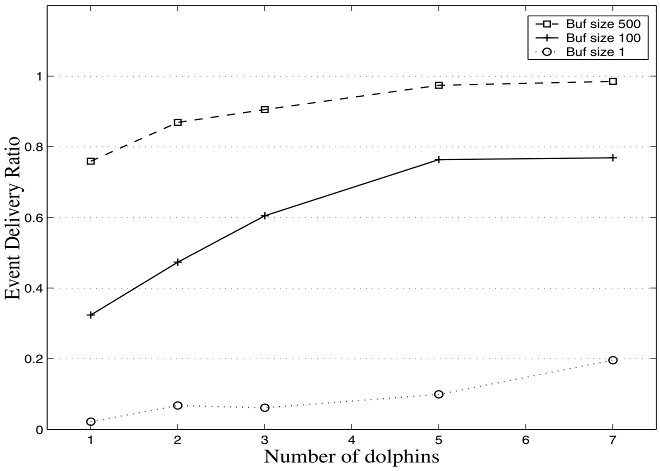

Fig. 1. Event Delivery Ratio as a function of number of dolphins

V. ANALYSIS

The simulation results obtained are a first step in this challenging research. We are also working on the development of an analytic model, to further support the conclusions in Section IV. First, we are trying to determine how the time needed to exhaustively harvesting the events depends on the number of agents. For easy of calculation, we considered a one dimensional approximation of the two-dimensional problem. Even if this seems an oversimplifying assumption, the ability of this model to catch essential principles has been advocated in [7]. Some "back of the envelope" calculations can be found in the longer technical report version of this paper [13].

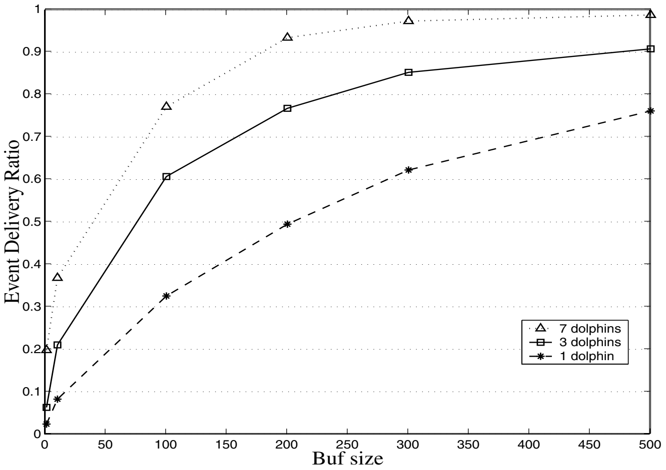

Fig. 2. Event Delivery Ratio as a function of Buf size

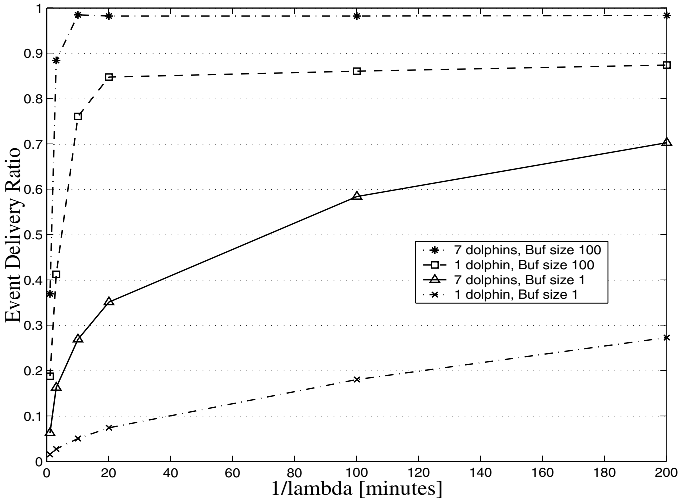

Fig. 3. Event Delivery Ratio as a function of the generation rate

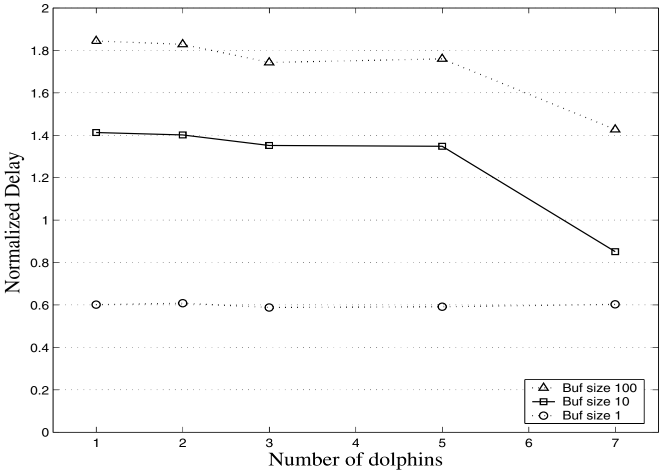

Fig. 4. Normalized delay for lambda=1 event every 2 minutes

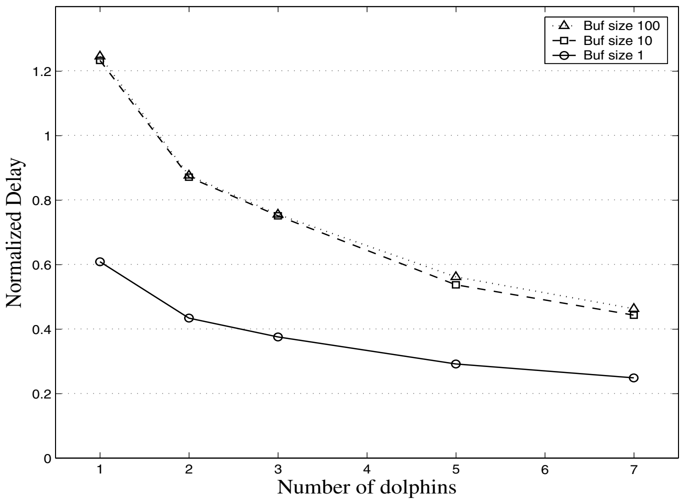

Fig. 5. Normalized delay for lambda=1 event every 200 minutes

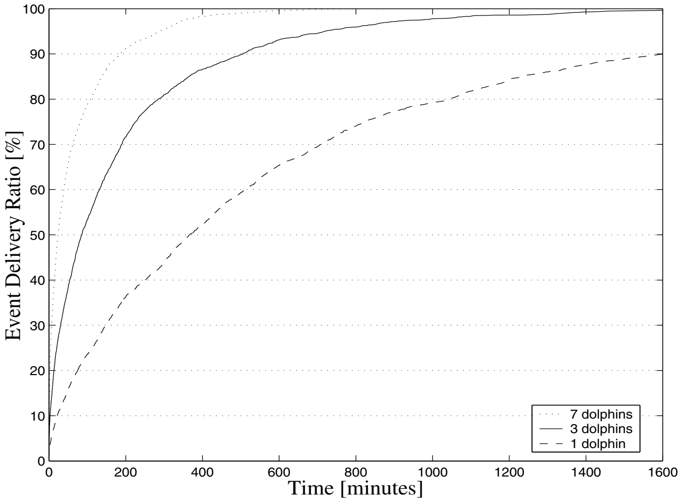

Fig. 6. Time evolution of Event Delivery Ratio as a function of the number of dolphins

VI. CONCLUSIONS

In this paper we proposed DDD, an approach to apply delay-tolerant networking in the underwater environment with extremely constrained communication resource, requiring energy-expensive transmission devices. In DDD, a few capable agents scour the field and collect information sensed by a number of distributed resource-limited devices. These sensors upload sensed events only when a dolphin is in direct one-hop range, to minimize the energy required for communications. Experimental results investigate how system parameters (e.g., the number of dolphins) should be tuned to obtain the desired average delay and ratio of harvested events, thus demonstrating the feasibility of the approach with very limited numbers of dolphins in most application scenarios. Our on-going research work mainly follows two directions. First, we are investigating more accurately the energy balance of different communication solutions (different modem choices, MAC layer protocols, 1-hop vs. multi-hop relaying). Second, we are continuing to work on the analytical model of the system, to obtain an expression of its main performance indicators.

REFERENCES

[1] I. Akyildiz, D. Pompili, and T. Melodia. Underwater acoustic sensor networks: research challenges. In *Elsevier Journal of Ad Hoc Networks*, volume 3, pages 257–281, Mar. 2005.

[2] S. Burleigh, K. Fall, V. Cerf, R. Durst, K. Scott, H. Weiss, L. Torgerson, and A. Hooke. Delay-Tolerant Networking: an Approach to Interplanetary Internet. In *IEEE Communications Magazine*, volume 41, pages 128–137, Jun. 2003.

[3] T. Clausen, P. Jacquet, and L. Viennot. Comparative study of cbr and tcp performance of manet routing protocols. In *IEEE MESA*, Jan. 2002. ETSI.

[4] J.-H. Cui, J. Kong, M. Gerla, and S. Zhou. Challenges: Building Scalable Mobile Underwater Wireless Sensor Networks for Aquatic Applications. In *IEEE Network*, volume 20, pages 12–18, May-Jun. 2006.

[5] K. Fall. A delay-tolerant network architecture for challenged internets. In *ACM SIGCOMM'03*, Aug. 2003.

[6] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball. The whoi micro-modem: An acoustic communications and navigation system for multiple platforms. In *IEEE/MTS OCEANS Conference*, Sept. 2005.

[7] A. Goel, S. Rai, and B. Krishnamachari. Sharp thresholds For monotone properties in random geometric graph. In *ACM STOC04*, Jun. 2004.

[8] J. Heidemann, Y. Li, A. Syed, J. Wills, and W. Ye. Underwater sensor networking: Research challenges and potential applications . In *USC/ISI Technical Report ISI-TR-2005*, 2005.

[9] R. Humston, D. B. Olson, and J. S. Ault. Behavioral Assumptions in Models of Fish Movement and Their Influence on Population Dynamics. In *Transactions of the America Fisheries Society*, volume 10, pages 1304–1328, Jun. 2004.

[10] D. B. Kilfoyle and A. B. Baggeroer. The State of the Art in Underwater Acoustic Telemetry. *IEEE Journal of Oceanic Engineering*, OE-25(1):4–27, January 2000.

[11] J.-Y. LeBoudec and M. Vojnovic. Perfect Simulation and Stationarity of a Class of Mobility Models. In *IEEE Infocom '05*, Mar. 2005.

[12] LinkQuest, Inc. Underwater Acoustic Modems. `http://www.link-quest.com/html/uwm_hr.pdf`.

[13] E. Magistretti, J. Kong, U. Lee, M. Gerla, and P. Bellavista. Mobile Delay-tolerant Approach to Long-term Energy-Efficient Underwater Sensor Networks . In *UCLA Technical Report*, 2006.

[14] Crossbow Mica 2 Motes. `http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_OEM_Edition_Datasheet.pdf`.

[15] S. PalChaudhuri, J.-Y. LeBoudec, and M. Vojnovic. Perfect Simulations of Random Trip Models. In *38th Annual Simulation Symposium*, Apr. 2005.

[16] J. Proakis, E. Sozer, J. Rice, and M. Stojanovic. Shallow Water Acoustic Networks. In *IEEE Communications Magazine*, volume 39, pages 114–119, Nov. 2001.

[17] The Qualnet Simulator. `http:://www.scalable-networks.com`.

[18] T. Small and Z. J. Haas. The Shared Wireless Infostation Model - A New Ad Hoc Networking Paradigm (or Where There is a Whale, There is a Way). In *ACM MOBIHOC'03*, Jun. 2003.

[19] L. F. M. Vieira, J. Kong, U. Lee, and M. Gerla. Analysis of Aloha Protocols for Underwater Acoustic Sensor Networks. In *ACM WUWNET'06 - Poster available at `http://wuwnet.engr.uconn.edu/wipposters/luiz_aloha.pdf`*, Sept. 2006.

[20] J. Wills, W. Ye, and J. Heidemann. Low-Power Acoustic Modem for Dense Underwater Sensor Networks. In *ACM WUWNET'06*, Sept. 2006.

[21] W. Zhao, M. Ammar, and E. Zegura. A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks. In *ACM MOBIHOC'04*, May 2004.